2. Übungsblatt

3.0 VU Datenmodellierung

21. November 2012

Allgemeines

In diesem Übungsteil sollten Sie Aufgabenstellungen aus den Bereich SQL und Normalformentheorie bearbeiten.

Lösen Sie die Beispiele eigenständig, denn in der Praxis (und bei der Prüfung) sind Sie auch auf sich alleine gestellt. Wir weisen Sie darauf hin, dass abgeschriebene Lösungen mit 0 Punkten beurteilt werden.

Geben Sie ein einziges PDF Dokument ab. Erstellen Sie Ihr Abgabedokument computerunterstützt. Wir akzeptieren keine gescannten handschriftlichen PDF-Dateien.

Deadlines

07.12. 06:55 Uhr	Upload über den CourseManager. Es werden keine verspäteten Abgaben akzeptiert.
15.12. 12:00 Uhr	Feedback im CourseManager verfügbar
16.12. 06:55 Uhr	Reservierung eines Termins für das Abgabegespräch

Abgabegespräch

1. Sie müssen sich über den CourseManager zu einem Abgabegespräch (=Prüfungsgespräch) anmelden. Bitte machen Sie das rechtzeitig, je später Sie sich anmelden, umso eingeschränkter ist das Terminangebot. Bis zur oben genannten Deadline garantieren wir Ihnen einen Termin.

2. Sie müssen mindestens einen Punkt auf das Übungsblatt bekommen. Wenn Sie weniger als einen Punkt auf das Blatt bekommen, oder kein Blatt abgegeben haben, sind Sie nicht zum Abgabegespräch zugelassen.

3. Sie kommen mit Ihrem Studierendenausweis zu der von Ihnen reservierten Zeit vorbei, und absolvieren das Abgabegespräch. Stoffgebiet des Abgabegesprächs sind die mit dem Übungsblatt abgedeckten Themengebiete. Wir setzen voraus, dass Sie sich mit Ihrer korrigierten Abgabe auseinandergesetzt haben.

4. Sie absolvieren Ihr Abgabegespräch gemeinsam mit anderen KollegInnen. Das Gespräch dauert ca. 60 Minuten.

5. Sie können auf die Abgabe maximal 15 Punkte erreichen. Diese setzen sich wie folgt zusammen:

   5 Punkte auf das Übungsblatt

   10 Punkte auf das Abgabegespräch

6. Die Assistenten tragen die Punkte des Abgabegesprächs in den CourseManager ein und Sie sehen dort, wieviele Punkte Sie bekommen haben.

7. Um die Lehrveranstaltung positiv abzuschließen, brauchen Sie mindestens 1 Punkt auf das Übungsblatt und 1 Punkt auf das Abgabegespräch.

8. Falls Sie nicht zu Ihrem Gesprächstermin erscheinen, bekommen Sie automatisch 0 Punkte und damit ein negatives Zeugnis.

SQL

Aufgabe 1 (eSQL) *[1.0 Punkte]*
Lösen Sie alle 10 (inkl. Unterpunkte) unter

`http://minteka.dbai.tuwien.ac.at/eSQL-tutorial/`

zur Verfügung gestellten SQL-Aufgaben des aktuellen Übungskurses. Loggen Sie sich dabei mit dem Usernamen und dem Passwort ein, das Sie bereits vom CourseManager kennen. Der Abschlusstest der Übung wird über dieselbe Plattform abgewickelt. Es empfiehlt sich daher zusätzlich auch mit Beispielen aus den vorigen Semestern zu üben.

Aufgabe 2 (Kaskadierendes Löschen) *[0.5 Punkte]*
Gegeben sei eine Datenbank mit den Relationen x, y und z, die wie folgt erstellt wurden:

```
CREATE TABLE x (
        xid INTEGER PRIMARY KEY,
        yid INTEGER REFERENCES y (yid) ON DELETE CASCADE
);

CREATE TABLE y (
        yid INTEGER PRIMARY KEY
);

CREATE TABLE z (
```

```
        zid INTEGER PRIMARY KEY,
        yid INTEGER REFERENCES y (yid) ON DELETE CASCADE
);
```

Die Ausprägungen der Relationen seien:

$$x = \{(1,1),(2,1),(3,3),(4,4),(5,3)\}$$
$$y = \{(1),(2),(3),(4),(5)\}$$
$$z = \{(1,1),(2,4),(3,3)\}$$

Geben Sie die Ausprägungen an, die sich durch das Ausführen der Statements

(a) `DELETE FROM` y `WHERE` yid = 3 und

(b) `DELETE FROM` y `WHERE` yid = 1 `OR` yid > 3

ergeben, jeweils auf die ursprüngliche Ausprägung angewendet.

Lösung:

(a) $x = \{(1,1),(2,1),(4,4)\},\ y = \{(1),(2),(4),(5)\},\ z = \{(1,1),(2,4)\}$

(b) $x = \{(3,3),(5,3)\},\ y = \{(2),(3)\},\ z = \{(3,3)\}$

Aufgabe 3 (Allquantifizierung) *[0.5 Punkte]*
In einem all-quantifizierten Ausdruck wird z.B.: nach jenen Bewerbern gesucht, die *alle* gespeicherten Programmiersprachen beherrschen. Diesen Sachverhalt kann man in klassischer Prädikatenlogik erster Stufe wie folgt ausdrücken:

$$\varphi(b) \equiv \mathit{Bewerber}(b) \land \forall p\ (\mathit{Programmiersprache}(p) \rightarrow \mathit{beherrscht}(b,p))$$

Erklären Sie die beiden Methoden, durch die ein Allquantor in SQL ausgedrückt werden kann, zunächst allgemein und dann auch an dem oben angeführten einfachen Beispiel. Geben Sie hierbei die entsprechenden Abfragen in SQL-Syntax an.

```
Bewerber(bkey)
Programmiersprache(pkey)
beherrscht(Bewerber.bkey, Programmiersprache.pkey)
```

Lösung:
Prinzipiell kann ein All-Quantor in SQL mit Hilfe von `COUNT` oder durch `NOT EXISTS`, `NOT IN` Konstrukt ausgedrückt werden.

Bei dem Ansatz mit `COUNT` werden zunächst alle Tupel in der betrachteten Domäne gezählt. Dann wird werden die Tupel in der betrachteten Domäne gezählt, die überdies die gewünschte Bedingung erfüllen. Stimmen die beiden Ergenisse überein, erfüllen alle betrachteten Tupel die Bedingung. Eine Lösung wäre daher:

```sql
SELECT b.bkey
FROM Bewerber b JOIN beherrscht p ON p.bkey = b.bkey
GROUP BY b.bkey
HAVING COUNT(*) = (SELECT COUNT(*) FROM Programmiersprache);
```

Der andere Ansatz nutzt die Äquivalenz $\forall x\ \varphi(x) \equiv \neg\exists x\ \neg\varphi(x)$. Wir wollen also all jene Bewerber finden, für die es keine Programmiersprache gibt, die sie nicht beherrschen.

```sql
SELECT b.bkey
FROM Bewerber b
WHERE not exists (SELECT * FROM Programmiersprache p
                  WHERE p.pkey NOT IN (SELECT a.pkey
                                       FROM beherrscht a
                                       WHERE a.bkey=b.bkey))
```

Normalformentheorie

Aufgabe 4 (Armstrong Axiome) *[0.5 Punkte]*
Gegeben ist ein Relationenschema $\mathcal{R} = \{ABCDE\}$ und eine Menge F von funktionalen Abhängigkeiten.

$$F = \{AC \to AE, E \to B, B \to AC, CE \to B\}$$

Betrachten Sie nun folgende beide "Axiome":

Wenn $\alpha \to \beta$, dann gilt $\beta \to \alpha$ (für $\alpha, \beta \subseteq \mathcal{R}$)	(Symmetrie)
Falls $\alpha, \beta \subseteq \mathcal{R}$, dann gilt $\alpha \to \alpha\beta$	(rechts-Verstärkung)

Diese beiden unsinnigen Axiome (Symmetrie und rechts-Verstärkung) folgen nicht aus den Armstrong Axiomen. Beweisen Sie dies, indem Sie zunächst zeigen, dass die Hülle F^+ von F unter Verwendung der Armstrong Axiome eine andere ist, als die Hülle F^+_{Sym} unter Verwendung von Armstrong Axiomen plus Symmetrie. Zeigen Sie weiters, dass F^+ ungleich der Hülle F^+_{r-V} unter Verwendung der Armstrong Axiome plus rechts-Verstärkung ist.

Lösung:
Wir zeigen, dass Symmetrie und rechts-Verstärkung (zusammen mit den Armstrong Axiomen) jeweils die Herleitung folgendes Axioms erlauben: Seien $\alpha, \beta \subseteq \mathcal{R}$, dann gilt $\alpha \to \beta$.

Armstrong Axiome plus Symmetrie: Seien $\alpha, \beta \subseteq \mathcal{R}$:

$$\cfrac{\cfrac{\cfrac{[\text{Refl.}]}{\mathcal{R} \to \alpha}}{\alpha \to \mathcal{R}}\ \text{Symmetrie}}{\alpha \to \beta}\ \text{Dekomp.}$$

Armstrong Axiome plus rechts-Verstärkung: Seien $\alpha, \beta \subseteq \mathcal{R}$:

$$\cfrac{\cfrac{\cfrac{}{\alpha \to \alpha}\,[\text{Refl.}]}{\alpha \to \alpha\beta}\,\text{rechts-Verstärkung}}{\alpha \to \beta}\,\text{Dekomp.}$$

Hiermit gilt dann, dass $F^+_{Sym} = \{\alpha \to \beta \mid \alpha, \beta \subseteq \mathcal{R}\}$, ebenso wie $F^+_{r-V} = F^+_{Sym}$. Da F keine FDs für das Attribut D enthält, gelten für D nur die trivialen FDs. F^+ enthält daher etwa nicht die FD $A \to D$, welche aber in F^+_{r-V} und F^+_{Sym} enthalten ist. Deshalb gilt $F^+ \neq F^+_{r-V}$ und $F^+ \neq F^+_{Sym}$.

Aufgabe 5 (Kanonische Überdeckung) *[0.5 Punkte]*
Bestimmen Sie die kanonische Überdeckung folgender Mengen funktionaler Abhängigkeiten über dem Relationenschema $ABCDEFG$ und dokumentieren Sie den Lösungsweg.

(a) $F^1 = \{AB \to CD,\ A \to D,\ D \to B,\ AD \to F,\ E \to DF,\ B \to F,\ D \to C\}$

(b) $F^2 = \{CD \to F,\ ADG \to B,\ E \to G,\ D \to CD,\ EG \to AE,\ G \to A,\ EFG \to B\}$

Lösung:

(a) $F^1_C = \{A \to D,\ B \to F,\ D \to BC,\ E \to D\}$

(b) $F^2_C = \{DG \to B,\ D \to CF,\ E \to G,\ EF \to B,\ G \to A\}$

Aufgabe 6 (Schlüsselbestimmung) *[0.5 Punkte]*

(a) Bestimmen Sie für folgendes Relationenschema samt funktionalen Abhängigkeiten alle Schlüssel und alle Superschlüssel.

$$\mathcal{R} = ABCDE$$
$$F = \{BC \to E,\ C \to B,\ A \to E,\ A \to D\}$$

Lösung:
AC ist der einzige Schlüssel. Die Menge der Superschlüssel:

$$\{AC, ABC, ACD, ACE, ABCD, ACDE, ABCE, ABCDE\}$$

(b) Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$\mathcal{R} = ABCDE$$
$$F = \{A \to BE, D \to C, A \to AC, E \to C\}$$

Erklären Sie, warum A und ACD keine Schlüssel sind.

Lösung:
A is kein Schlüssel, da $D \notin \mathsf{AttrHülle}(F, \{A\})$. Da nun aber AD ein Schlüssel ist, kann ACD nicht minimal sein und ist deshalb kein Schlüssel.

Aufgabe 7 (Synthesealgorithmus) *[0.5 Punkte]*
Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$\mathcal{R} = ABCDEFG$$
$$F = \{B \to AC,\ BCD \to G,\ BDF \to C,\ A \to E,\ C \to A,\ G \to F\}$$

Gesucht ist eine verlustlose und abhängigkeitserhaltende Zerlegung in dritter Normalform. Wenden Sie hierzu den Synthesealgorithmus an und dokumentieren Sie das Ergebnis der einzelnen Schritte. Bestimmen Sie alle Schlüssel von $\mathcal{R}$ und allen Relationen der Zerlegung.

Lösung:

1. Bestimmung der kanonischen Überdeckung:

$$F_c = \{B \to C,\ BD \to G,\ A \to E,\ C \to A,\ G \to F\}$$

2. Erstelle Relationenschemata für jedes Element von F_c:

Relationenschema	Geltende FDs
$\mathcal{R}_1 = BC$	$F_1 = \{B \to C\}$
$\mathcal{R}_2 = BDG$	$F_2 = \{BD \to G\}$
$\mathcal{R}_3 = AE$	$F_3 = \{A \to E\}$
$\mathcal{R}_4 = CA$	$F_4 = \{C \to A\}$
$\mathcal{R}_5 = GF$	$F_5 = \{G \to F\}$

3. Bestimmung aller Kandidatenschlüssel von $\mathcal{R}$ bzgl. F_c: BD.

   BD ist in $\mathcal{R}_2$ der erzeugten Teilschemata enthalten.

4. Kein Relationenschema ist zu eliminieren.

Ergebnis (Schlüssel sind unterstrichen):

Relationenschema	Geltende FDs
$\mathcal{R}_1 = \underline{B}C$	$F_1 = \{B \to C\}$
$\mathcal{R}_2 = \underline{BD}G$	$F_2 = \{BD \to G\}$
$\mathcal{R}_3 = \underline{A}E$	$F_3 = \{A \to E\}$
$\mathcal{R}_4 = \underline{C}A$	$F_4 = \{C \to A\}$
$\mathcal{R}_5 = \underline{G}F$	$F_5 = \{G \to F\}$

Aufgabe 8 (Normalformen) *[0.5 Punkte]*
Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$\mathcal{R} = ABCDEFG$$
$$F = \{CEG \to BA,\ A \to CDF,\ F \to EG\}$$

Geben Sie an, ob $\mathcal{R}$

(a) in dritter Normalform ist,

(b) in Boyce-Codd-Normalform ist,

und begründen Sie Ihre Antworten.

Lösung:
A, CEG und CF sind die einzigen Schlüssel von $\mathcal{R}$. $\mathcal{R}$ ist in dritter Normalform. Allerdings ist $F \to E$ weder trivial, noch ist F Superschlüssel von $\mathcal{R}$. Daher ist $\mathcal{R}$ nicht in BCNF.

Aufgabe 9 (Dekompositionsalgorithmus) *[0.5 Punkte]*
Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$R = ABCD$$
$$F = \{AC \to B,\ AB \to D\}$$

Gesucht ist eine verlustlose Zerlegung in Boyce-Codd-Normalform. Wenden Sie hierzu den Dekompositionsalgorithmus an und dokumentieren Sie das Ergebnis der einzelnen Schritte. Bestimmen Sie alle Schlüssel von $\mathcal{R}$ und allen Relationen der Zerlegung. Ist die Zerlegung abhängigkeitserhaltend? Wenn die Zerlegung nicht abhängigkeitserhaltend ist, geben Sie an, welche Abhängigkeiten verloren gegangen sind.

Lösung:
Der einzige Schlüssel von $\mathcal{R}$ ist AC. Die FD $AB \to D$ verletzt die BCNF, da sie nicht trivial und AB kein Superschlüssel ist. Zerlege $\mathcal{R}$ daher in:

$\mathcal{R}_1 = ABD$	$F_1 = \{AB \to D\}$	Schlüssel: AB
$\mathcal{R}_2 = ABC$	$F_2 = \{AC \to B\}$	Schlüssel: AC

$\mathcal{R}_1$ und $\mathcal{R}_2$ sind beide in BCNF und die Zerlegung ist abhängigkeitserhaltend, da bereits $F = F_1 \cup F_2$ gilt.